

Creating and using classes in JavaScript

JavaScript is not really an OOP programming language (**based on classes and objects**, like C++, or PHP), it is a scripting language (based on functions) that is executed in browser. But, you can also create "classes" in JavaScript, that can be the base for an Object (defined with the "new" keyword). A **class in JavaScript** is created with the special word: function , using this syntax:

```
className = function() {  
    // code of the className class  
}
```

A class can contain public and private variables (called also properties) and functions (also called methods).

The private variables, and functions are defined with the keyword "var". They can be used only inside the class body.

The public variables, and functions are defined with the special word: "this". They can be used inside, and outside the class body.

The complete syntax to create a class in JavaScript:

```
className = function() {  
    var somePrivateVariable = 'private_value';  
  
    var somePrivateFunction = function(/* parameters */) {  
        // some code  
    }  
  
    this.somePublicVariable = 'public_value';  
  
    this.somePublicFunction = function(/* parameters */) {  
        // code ...  
    }  
}
```

Then the code of the script using the "class" object could be like this:

```
myObject = new className();
```

Accessing properties and functions of the class

Once the class is created, and it is set an object of that class, you can use /call the public properties (variables) and functions defined in that class, using the name of the object.

Syntax:

```
myObject = new className();  
  
// sets value of a public property  
myObject.somePublicVariable = 'value';  
  
// calls a public function of the class  
var method = myObject.somePublicFunction(/* parameters */);
```

- The public variables and functions created in the class (with the "this" keyword) can be accessed

inside class body using the "this" keyword (as they was defined).
Here is a concrete example (see the comments in code).

```
<script type="text/javascript"><!--
// From - http://coursesweb.net/javascript/
// create the class
getSum = function() {
    var nr1 = 0;          // private
    this.nr2 = 0;         // public

    // private function to set the value of nr1 (which is private)
    var setNr1 = function(nr) {
        nr1 = nr;
    }

    // public function
    // returns the sum of nr1 and nr2 (using "this." because it was defined
    // with this keyword)
    this.sum = function(nr) {
        setNr1(nr);      // calls the private setNr1() function to set the
        // value of nr1

        return nr1 + this.nr2;
    }
}

// sets an object of getSum class
var obSum = new getSum();

// accesses the public property
alert(obSum.nr2);           // 0

// sets another value for nr2, and gets the sum
obSum.nr2 = 6;
var sum64 = obSum.sum(4);
alert(sum64);               // 10

// test to see if nr1 (private) can be accessed
alert(obSum.nr1);           // undefined
//-->
</script>
```

- You can add parameters in the class definition, so, when it is defined an object of that class, it must be added a value for that parameter.

Syntax:

```
className = function(parameter) {
    // code of the className class
}

myObject = new className('value');
```

Here is another example, with a class that requires a parameter.

```
<script type="text/javascript"><!--
// From - http://coursesweb.net/javascript/
// create getSum class with a parameter
```

```
getSum = function(nr) {
    var nr1 = 0;
    this.nr2 = 0;

    // sets the value of nr1 property
    var setNr1 = function(nr) {
        nr1 = nr;
    }

    // returns the sum of nr1 (which is private) and nr2 (using "this."
    because it was defined with this keyword)
    this.sum = function() {
        return nr1 + this.nr2;
    }

    setNr1(nr);      // calls the setNr1() function when an object of this
    class is created
}

// sets an object of getSum class
var obSum = new getSum(3);
alert(obSum.nr2);          // 0

// sets another value for nr2, and gets the sum
obSum.nr2 = 5;
var sum35 = obSum.sum();
alert(sum35);              // 8
//-->
</script>
```